

ACCOUNTING FOR LINK UTILIZATION IN SCHEDULING AND BILLING

FIELD OF THE INVENTION

The present invention is related to transferring data in a telecommunications network. More specifically, the present invention is related to transferring data in a telecommunications network by determining a link length for the packet that will be sent into the network based on the memory length and at least one parameter, where the link length correctly corresponds to the packet's physical line bandwidth usage in the network.

BACKGROUND OF THE INVENTION

In high end packet switches buffer management, scheduling, and billing functions will most likely be disconnected from any encapsulation, and framing functions, i.e. these two different sets of functions will most likely be implemented in different sub-components. In this scenario, a simple back-pressure based approach of running Scheduler slightly faster and then backpressuring it to account for any encapsulation and framer overhead downstream will not give accurate scheduling. Billing information will also not be accurate without accounting for encapsulation and framing overhead downstream.

In the present invention, expressions are given to accurately account for any packet size changes to the downstream of the Scheduling and Billing functions to provide accurate scheduling and billing as per packet size sent on the physical line.

The present invention gives accurate scheduling and billing information as per packet size (bandwidth utilization) on the physical line, instead of basing schedule rates and billing information as per packet size that is stored in the switch
5 internal memory. Also, with programmable parameters downstream packet size changes can be accounted for due to different encapsulation and framing methodologies.

SUMMARY OF THE INVENTION

The present invention pertains to an apparatus for
10 transferring data in a telecommunications network. The apparatus comprises a memory in which a packet memory length is stored. The apparatus comprises means for determining a link length for the packet that will be sent into the network based on the memory length and at least one parameter, where the link length correctly
15 corresponds to the packet's physical line bandwidth usage in the network. The apparatus comprises means for sending the packet having the link length to the network.

The present invention pertains to a method for transferring data in a telecommunications network. The method
20 comprises the steps of storing in a memory a memory length of a packet. There is the step of determining a link length for the packet that will be sent into the network based on the memory length and at least one parameter, where the link length correctly corresponds to the packet's physical line bandwidth usage in the
25 network. There is the step of sending the packet having the link length to the network.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, the preferred embodiment of the invention and preferred methods of practicing the invention are illustrated in which:

5 Figure 1 is a schematic representation of the present invention.

Figure 2 is a block diagram of the present invention.

DETAILED DESCRIPTION

Referring now to the drawings wherein like reference
10 numerals refer to similar or identical parts throughout the several views, and more specifically to figures 1 and 2 thereof, there is shown an apparatus 10 for transferring data in a telecommunications network 16. The apparatus 10 comprises a memory 12 in which a packet memory length is stored. The apparatus 10 comprises means
15 for determining a link length for the packet that will be sent into the network 16 based on the memory length and at least one parameter, where the link length correctly corresponds to the packet's physical line bandwidth usage in the network 16. The apparatus 10 comprises means for sending the packet having the link
20 length to the network 16.

Preferably, the apparatus 10 includes a scheduler 20 for scheduling when the packet having the link length is to be sent into the network 16 based on the link length of the packet. The determining means 14 preferably includes a controller 22.

Preferably, the controller 22 includes a billing module 24 for billing for the physical line bandwidth usage by the packet based on the link length of the packet.

The determining means 14 preferably determines the link
5 length according to

$$LinkLength = \left\lceil \frac{\{MemoryLength - HeaderSize + MPLSAdj\}}{FragmentSize} \right\rceil \times \{HeaderSize + FrameOverhead\} + \{MemoryLength - HeaderSize + MPLSAdj\} + LastFragmentPad$$

Where,

LinkLength	=	Length of the packet as seen on the physical line
10 MemoryLength	=	Length of the packet as stored in the internal memory (= Header + Payload)
HeaderSize	=	Length of the Header for this packet
MPLSAdj	=	MPLS label push (positive value) / pop (negative value) amount
15 FragmentSize	=	Required payload size of each packet fragment
FrameOverhead	=	Number of bytes that will be added to each frame/packet in SONET/SDH framing
LastFragmentPad	=	Number of bytes that need be added to the last fragment of a fragmented packet to make the
20		fragment minimum size

Preferably, the link length is determined by the controller 22 if the packet is an ATM cell according to

$$\text{LinkLength} = \text{MemoryLength} + \text{FrameOverhead}.$$

The apparatus 10 preferably includes a port card 26 having the memory 12, determining means 14, sending means 18, scheduler 20 and billing module 24. Preferably, the memory 12 includes a packet storage handler 32 in communication with the billing module 24 and the scheduler 20. The sending means 18 preferably includes a packet framing and line interface 30 in communication with the packet storage handler 32.

Preferably, the sending means 18 includes a fragmentation/segmentation/encapsulation/decapsulation module 34 in communication with the packet storage handler 32 and the packet framing and line interface 30.

The controller 22 preferably frees the memory 12 based on the memory length of the packet that has been sent into the network 16. Preferably, the controller 22 determines memory length according to

$$\text{MemoryLength} = \text{LinkLength} + \text{HeaderSize} + \text{MPLSAdj} - \text{LastFragmentpad} - [\text{NumFragments} \times (\text{HeaderSize} + \text{FrameOverhead})]$$

Where

$$\text{NumFragments} = \left\lceil \frac{\text{LinkLength}}{(\text{HeaderSize} + \text{FragmentSize} + \text{FrameOverhead})} \right\rceil$$

The present invention pertains to a method for transferring data in a telecommunications network 16. The method comprises the steps of storing in a memory 12 a memory length of a packet. There is the step of determining a link length for the packet that will be sent into the network 16 based on the memory length and at least one parameter, where the link length correctly corresponds to the packet's physical line bandwidth usage in the network 16. There is the step of sending the packet having the link length to the network 16.

Preferably, there is the step of scheduling when the packet having the link length is to be sent into the network 16 based on the link length of the packet. There is preferably the step of billing for the physical line bandwidth usage by the packet based on the link length of the packet. Preferably, the determining step includes the step of determining the link length according to

$$\text{LinkLength} = \left\lceil \frac{\{\text{MemoryLength} - \text{HeaderSize} \pm \text{MPLSAdj}\}}{\text{FragmentSize}} \right\rceil \times \{\text{HeaderSize} + \text{FrameOverhead}\} + \{\text{MemoryLength} - \text{HeaderSize} \pm \text{MPLSAdj}\} + \text{LastFragmentPad}$$

Where,

LinkLength	=	Length of the packet as seen on the physical line
MemoryLength	=	Length of the packet as stored in the internal memory (= Header + Payload)
HeaderSize	=	Length of the Header for this packet

MPLSAdj = MPLS label push (positive value) / pop
(negative value) amount

FragmentSize = Required payload size of each packet fragment

5 FrameOverhead = Number of bytes that will be added to each
frame/packet in SONET/SDH framing

LastFragmentPad = Number of bytes that need be added to the last
fragment of a fragmented packet to make the
fragment minimum size.

10 The determining step preferably includes the step of
determining the link length for an ATM cell according to

$$\text{LinkLength} = \text{MemoryLength} + \text{FrameOverhead}.$$

Preferably, the sending step includes the step of sending
the packet to the network 16 through a port card 26 having the
15 memory 12. There is preferably the step of freeing the memory 12
based on the memory length of the packet that has been sent into
the network 16. Preferably, the freeing step includes the step of
determining memory length according to

$$\text{MemoryLength} = \text{LinkLength} + \text{HeaderSize} \mp \text{MPLSAdj} - \text{LastFragmentpad} \\ - [\text{NumFragments} \times (\text{HeaderSize} + \text{FrameOverhead})]$$

20 Where,

$$NumFragments = \left\lceil \frac{LinkLength}{(HeaderSize + FragmentSize + FrameOverhead)} \right\rceil$$

In the operation of the invention, in packet based switch port cards due to MPLS (Multi Protocol Label Switching) label push/pops, Ipv4 (Internet Protocol version 4) encapsulation of Ipv6
5 packets, packet fragmentation into smaller packets, packet segmentation into ATM (Asynchronous Transfer Mode) cells, SONET/SDH (Synchronous Optical Network/Synchronous Digital Hierarchy) framing overhead, and/or any other reason packet "link length" (length of the packet as seen on the physical link) is different than the
10 packet "memory length" (length of the packet as stored in port card internal memory); when Buffer management (handling of packet storage buffers, and usage billing counters), packet Scheduling (sending out packets stored in internal memory at appropriate rates for different flows), and packet processing (MPLS label operations,
15 framing etc) are happening in different sub-components of the port card.

In the apparatus, Buffer management, Scheduling, and cell/packet processing (segmentation, encapsulation, framing etc.) are implemented in separate components. A cell or packet comes
20 into the system with a Header and Payload, which gets stored into the port card internal memory. This amount (Header + payload size) is termed as the "memory length" of that packet as that is what gets stored in the memory. Before the cell/packet is sent out onto the line different operations (MPLS label push/pops, packet
25 fragmentation, packet segmentation into ATM cells, Ipv4 encapsulation of Ipv6 packet, SONET/SDH framing etc.,) are

performed in the cell/packet processing components. To make the Scheduler and Billing modules account for the difference in packet length seen on the physical line versus the packet size stored in the internal memory, a few parameters are stored per each output
5 port and per each flow, and based on these parameters and packet "memory length" packet "link length" will be calculated, and that is used in Scheduler and Billing modules. By using packet "link length" in Scheduler and Billing modules physical line bandwidth usage is correctly scheduled, and billed for.

10 Following expression calculates packet "link length" from packet "memory length", and output port and flow based information programmed into the port card. All values are in terms of number of bytes.

$$\text{LinkLength} = \left\lceil \frac{\{\text{MemoryLength} - \text{HeaderSize} \pm \text{MPLSAdj}\}}{\text{FragmentSize}} \right\rceil \times \{\text{HeaderSize} + \text{FrameOverhead}\} + \{\text{MemoryLength} - \text{HeaderSize} \pm \text{MPLSAdj}\} + \text{LastFragmentPad}$$

15 Where,

LinkLength	=	Length of the packet as seen on the physical line
MemoryLength	=	Length of the packet as stored in the internal memory (= Header + Payload)
20 HeaderSize	=	Length of the Header for this packet
MPLSAdj	=	MPLS label push (positive value) / pop (negative value) amount
FragmentSize	=	Required payload size of each packet fragment

FrameOverhead = Number of bytes that will be added to each
frame/packet in SONET/SDH framing

LastFragmentPad= Number of bytes that need be added to the last
fragment of a fragmented packet to make the
5 fragment minimum size

For an IPv6 packet coming in on a flow with "IPv4 encapsulation of
IPv6" enabled, MemoryLength represents PDU (Payload Data Unit) as
IPv6 header becomes part of the payload when IPv4 header is added
on the top. Also, if the incoming datagram is an ATM cell, then
10 the above expression will be simplified as follows:

$$\text{LinkLength} = \text{MemoryLength} + \text{FrameOverhead}.$$

Once the packet is scheduled out of the internal memory
and billing counts are updated, memory used by that packets need to
be freed up. Since, only "memory length" amount of bytes for that
15 packet were stored into the memory as the packet came into the port
card, the same number of bytes need to be freed from the memory.
For this purpose, packet "memory length" needs to be re-calculated
from the above computed packet "link length". The following
expression calculates packet "memory length" from the packet "link
20 length", and output port and flow based information programmed into
the port card.

$$\text{MemoryLength} = \text{LinkLength} + \text{HeaderSize} + \text{MPLSAdj} - \text{LastFragmentpad} \\ - [\text{NumFragments} \times (\text{HeaderSize} + \text{FrameOverhead})]$$

Where,

$$NumFragments = \left\lceil \frac{LinkLength}{(HeaderSize + FragmentSize + FrameOverhead)} \right\rceil.$$

The Packet Link length is calculated in the BufferManagement module and packet Memory length is calculated in the Scheduler.

5 Although the invention has been described in detail in
the foregoing embodiments for the purpose of illustration, it is to
be understood that such detail is solely for that purpose and that
variations can be made therein by those skilled in the art without
departing from the spirit and scope of the invention except as it
10 may be described by the following claims.